FIG. 1

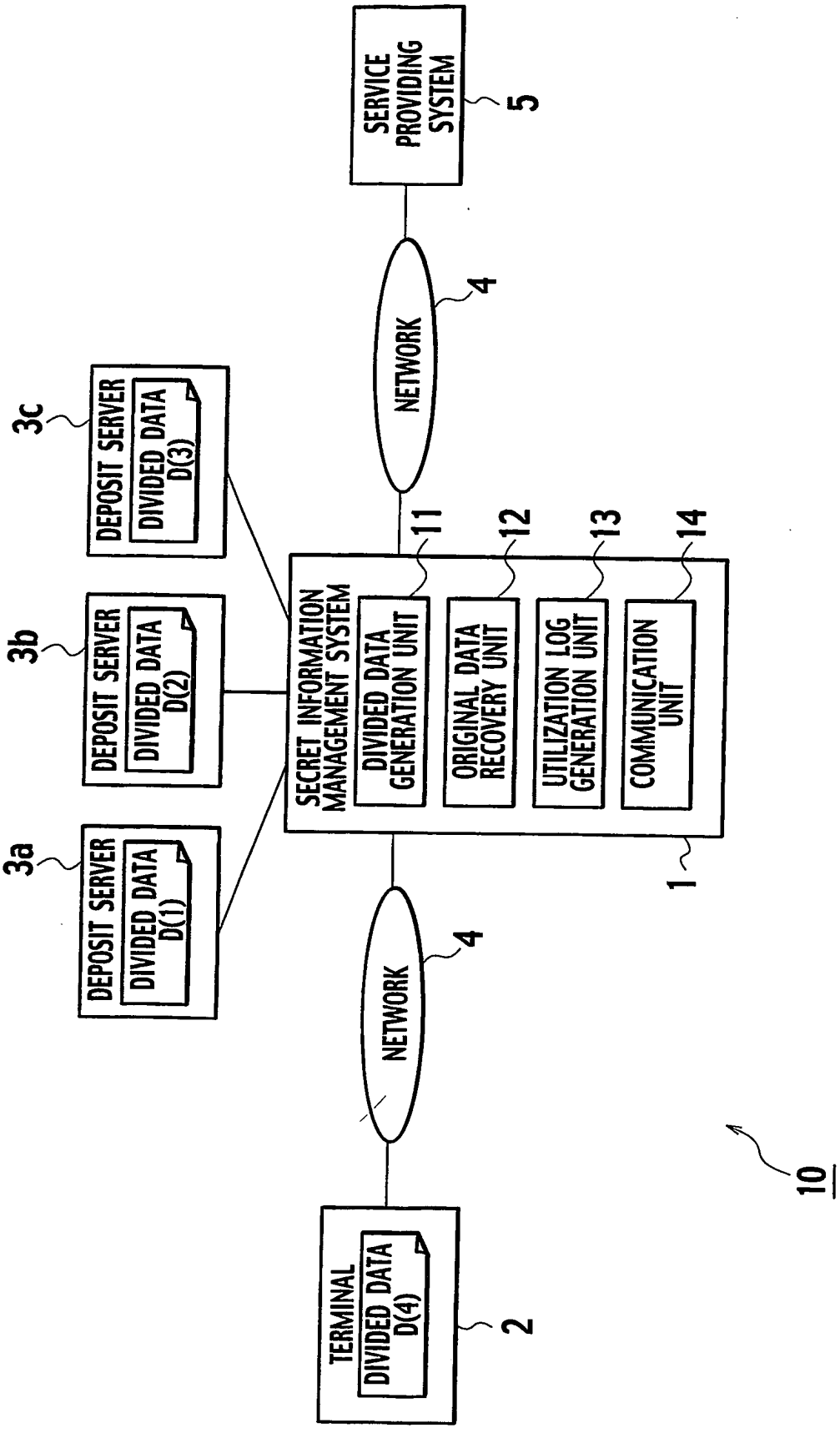
SERVICE PROVIDING SYSTEM
5
NETWORK
4
3c
DEPOSIT SERVER
DIVIDED DATA D(3)
3b
DEPOSIT SERVER
DIVIDED DATA D(2)
3a
DEPOSIT SERVER
DIVIDED DATA D(1)
SECRET INFORMATION MANAGEMENT SYSTEM
11
DIVIDED DATA GENERATION UNIT
12
ORIGINAL DATA RECOVERY UNIT
13
UTILIZATION LOG GENERATION UNIT
14
COMMUNICATION UNIT
1
NETWORK
4
TERMINAL
DIVIDED DATA D(4)
2
10

FIG. 2

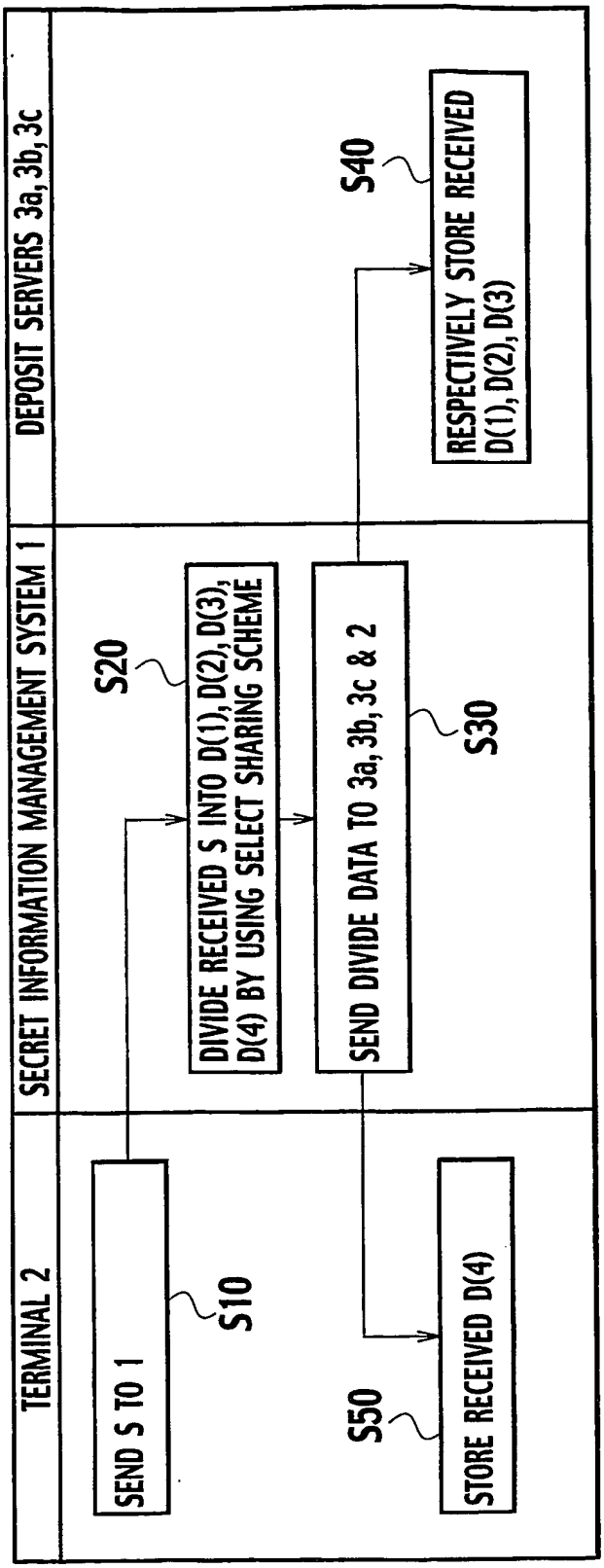
TERMINAL 2
SECRET INFORMATION MANAGEMENT SYSTEM 1
DEPOSIT SERVERS 3a, 3b, 3c
SEND S TO 1
S10
S20
DIVIDE RECEIVED S INTO D(1), D(2), D(3), D(4) BY USING SELECT SHARING SCHEME
SEND DIVIDE DATA TO 3a, 3b, 3c & 2
S30
S40
RESPECTIVELY STORE RECEIVED D(1), D(2), D(3)
S50
STORE RECEIVED D(4)

FIG. 3

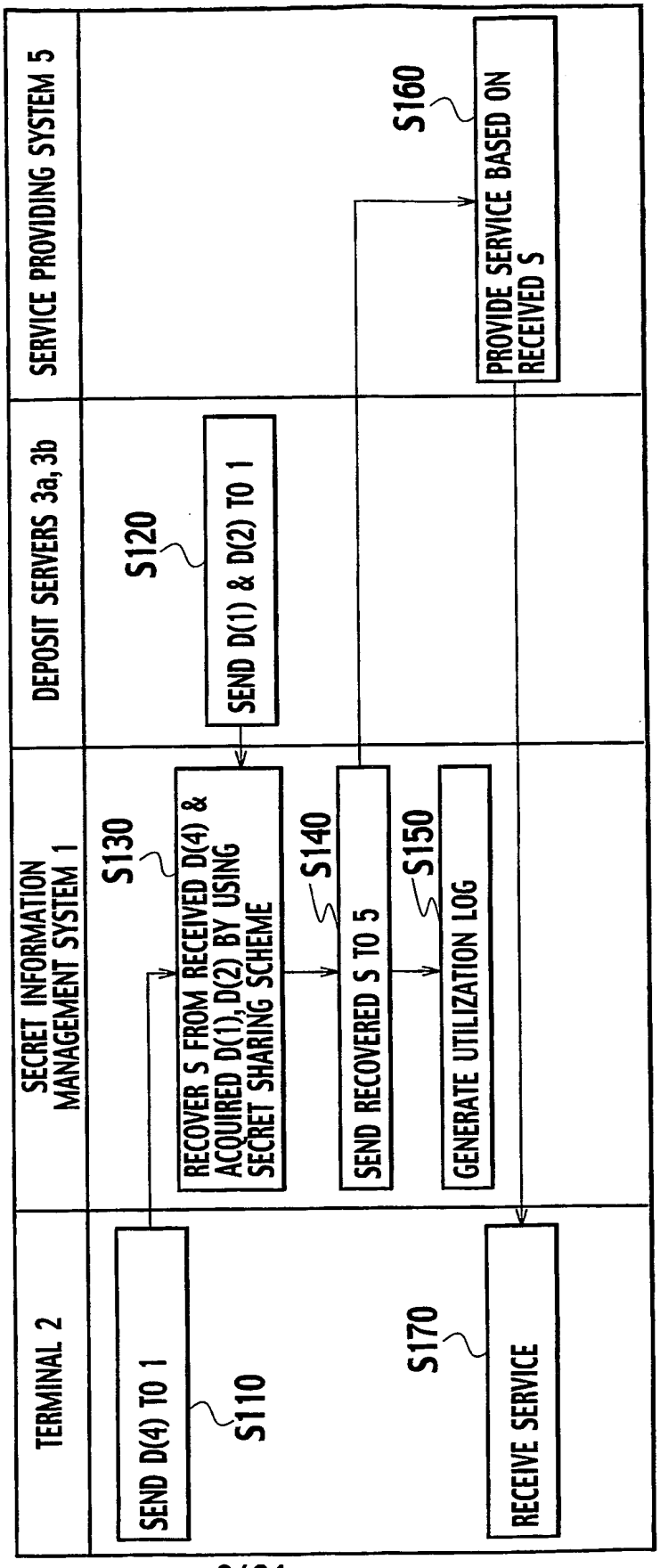
TERMINAL 2
SECRET INFORMATION MANAGEMENT SYSTEM 1
DEPOSIT SERVERS 3a, 3b
SERVICE PROVIDING SYSTEM 5
SEND D(4) TO 1
S110
SEND D(1) & D(2) TO 1
S120
RECOVER S FROM RECEIVED D(4) & ACQUIRED D(1), D(2) BY USING SECRET SHARING SCHEME
S130
SEND RECOVERED S TO 5
S140
GENERATE UTILIZATION LOG
S150
PROVIDE SERVICE BASED ON RECEIVED S
S160
RECEIVE SERVICE
S170

FIG. 4

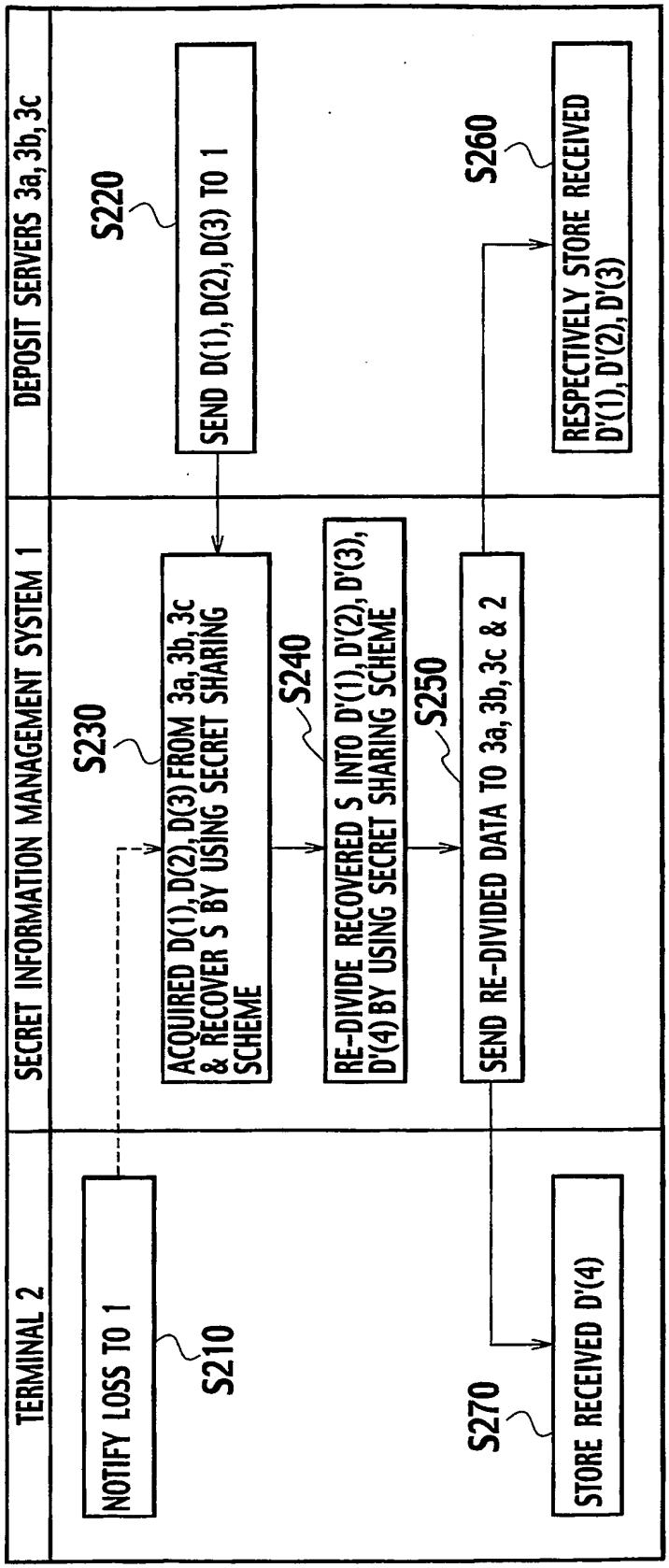
TERMINAL 2
SECRET INFORMATION MANAGEMENT SYSTEM 1
DEPOSIT SERVERS 3a, 3b, 3c
NOTIFY LOSS TO 1
S210
S220
SEND D(1), D(2), D(3) TO 1
S230
ACQUIRED D(1), D(2), D(3) FROM 3a, 3b, 3c & RECOVER S BY USING SECRET SHARING SCHEME
S240
RE-DIVIDE RECOVERED S INTO D'(1), D'(2), D'(3), D'(4) BY USING SECRET SHARING SCHEME
S250
SEND RE-DIVIDED DATA TO 3a, 3b, 3c & 2
S260
RESPECTIVELY STORE RECEIVED D'(1), D'(2), D'(3)
S270
STORE RECEIVED D'(4)

FIG. 5

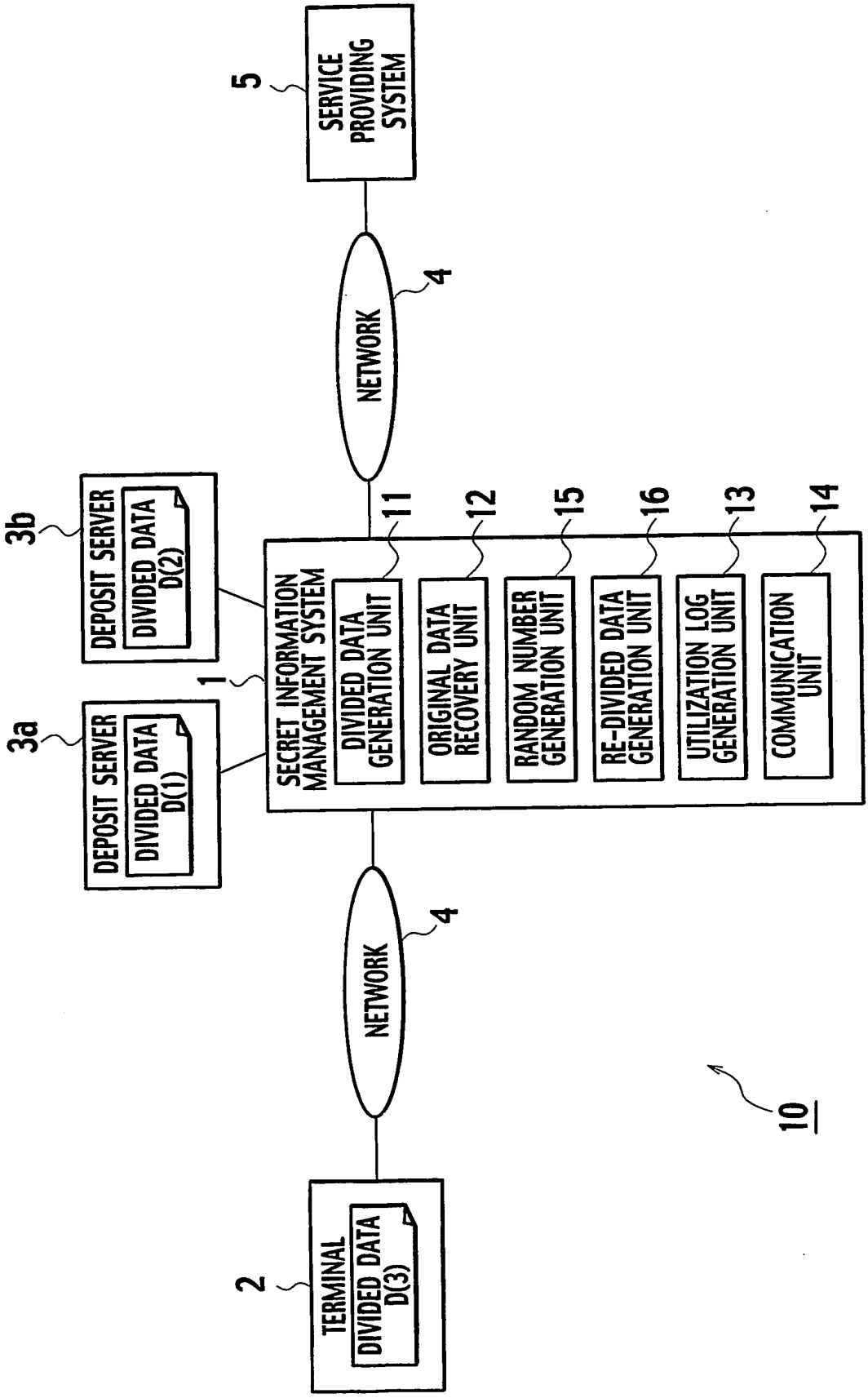
5
SERVICE PROVIDING SYSTEM
4
NETWORK
3b
DEPOSIT SERVER
DIVIDED DATA D(2)
3a
DEPOSIT SERVER
DIVIDED DATA D(1)
1
SECRET INFORMATION MANAGEMENT SYSTEM
11
DIVIDED DATA GENERATION UNIT
12
ORIGINAL DATA RECOVERY UNIT
15
RANDOM NUMBER GENERATION UNIT
16
RE-DIVIDED DATA GENERATION UNIT
13
UTILIZATION LOG GENERATION UNIT
14
COMMUNICATION UNIT
4
NETWORK
2
TERMINAL
DIVIDED DATA D(3)
10

FIG. 6

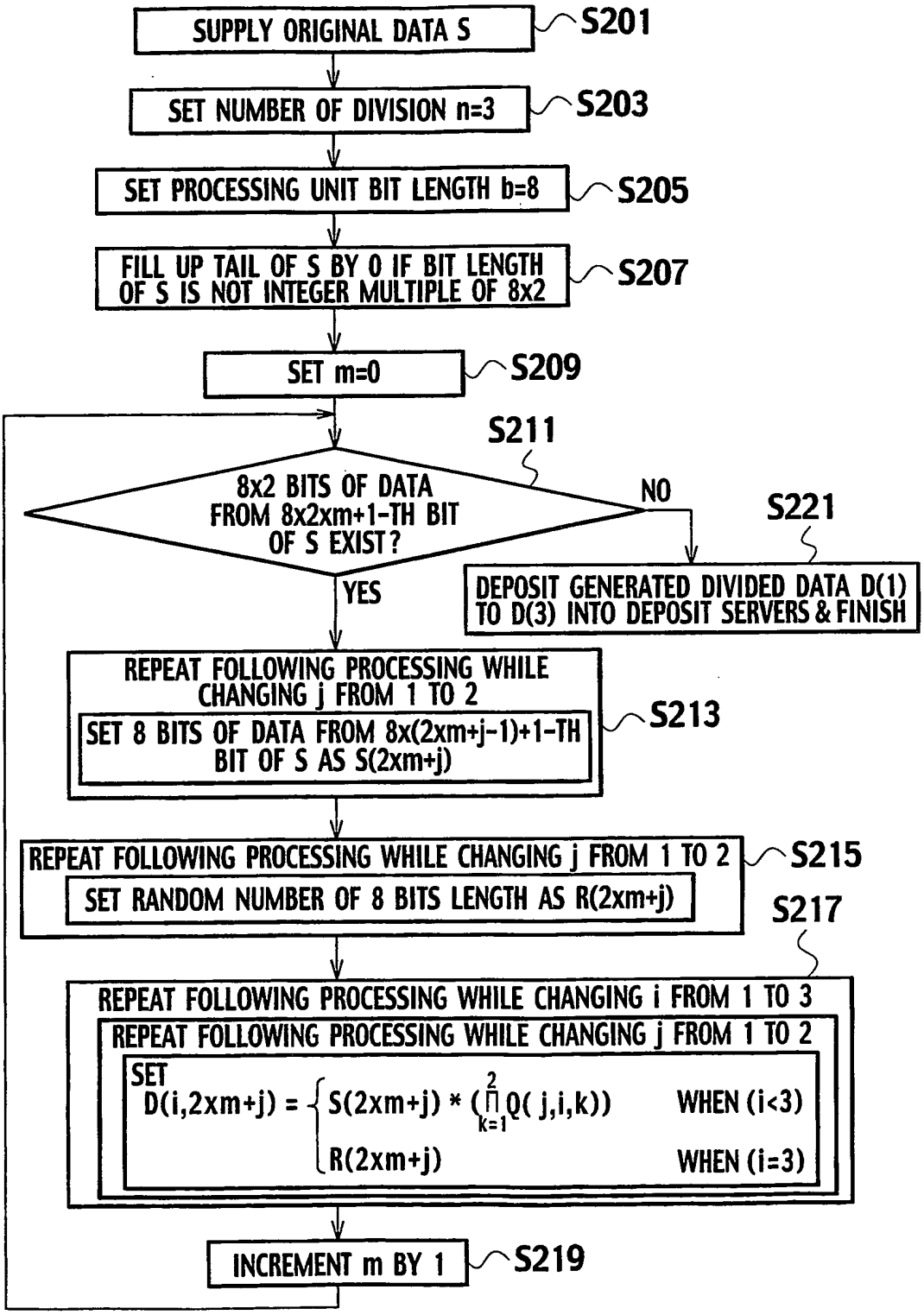
SUPPLY ORIGINAL DATA S
S201
SET NUMBER OF DIVISION n=3
S203
SET PROCESSING UNIT BIT LENGTH b=8
S205
FILL UP TAIL OF S BY 0 IF BIT LENGTH OF S IS NOT INTEGER MULTIPLE OF 8x2
S207
SET m=0
S209
S211
8x2 BITS OF DATA FROM 8x2xm+1-TH BIT OF S EXIST?
NO
S221
DEPOSIT GENERATED DIVIDED DATA D(1) TO D(3) INTO DEPOSIT SERVERS & FINISH
YES
REPEAT FOLLOWING PROCESSING WHILE CHANGING j FROM 1 TO 2
SET 8 BITS OF DATA FROM 8x(2xm+j-1)+1-TH BIT OF S AS S(2xm+j)
S213
REPEAT FOLLOWING PROCESSING WHILE CHANGING j FROM 1 TO 2
SET RANDOM NUMBER OF 8 BITS LENGTH AS R(2xm+j)
S215
S217
REPEAT FOLLOWING PROCESSING WHILE CHANGING i FROM 1 TO 3
REPEAT FOLLOWING PROCESSING WHILE CHANGING j FROM 1 TO 2
SET
D(i,2xm+j) = S(2xm+j) * (∏ k=1 to 2 Q(j,i,k)) WHEN (i<3)
R(2xm+j) WHEN (i=3)
INCREMENT m BY 1
S219

FIG. 7

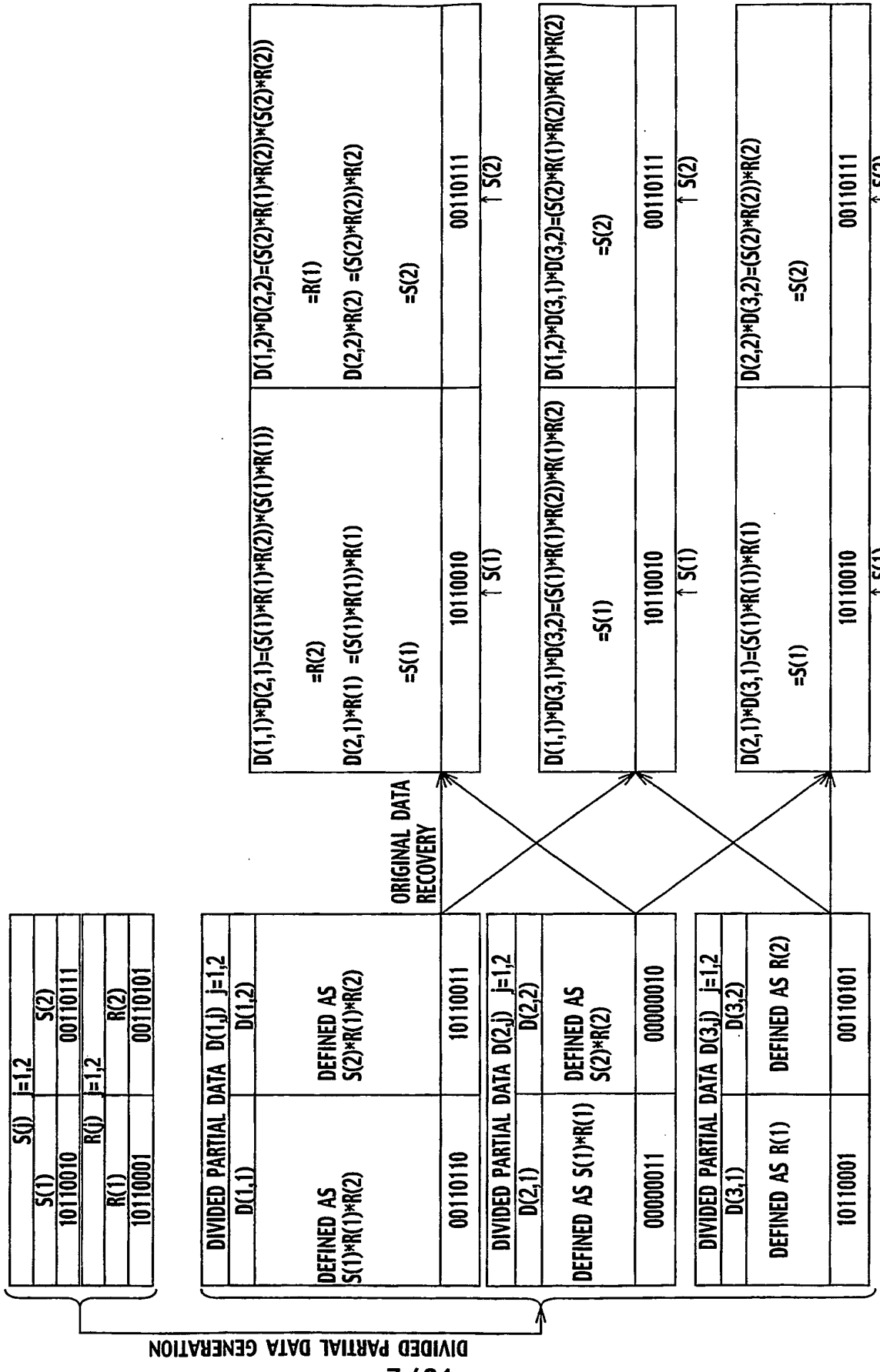
S(j) j=1,2
S(1)
10110010
S(2)
00110111
R(j) j=1,2
R(1)
10110001
R(2)
00110101
DIVIDED PARTIAL DATA GENERATION
DIVIDED PARTIAL DATA D(1,j) j=1,2
D(1,1)
DEFINED AS S(1)*R(1)*R(2)
D(1,2)
DEFINED AS S(2)*R(1)*R(2)
00110110
10110011
DIVIDED PARTIAL DATA D(2,j) j=1,2
D(2,1)
DEFINED AS S(1)*R(1)
D(2,2)
DEFINED AS S(2)*R(2)
00000011
00000010
DIVIDED PARTIAL DATA D(3,j) j=1,2
D(3,1)
DEFINED AS R(1)
D(3,2)
DEFINED AS R(2)
10110001
00110101
ORIGINAL DATA RECOVERY
D(1,1)*D(2,1)=(S(1)*R(1)*R(2))*(S(1)*R(1))
=R(2)
D(2,1)*R(1) =(S(1)*R(1))*R(1)
=S(1)
10110010
↑ S(1)
D(1,2)*D(2,2)=(S(2)*R(1)*R(2))*(S(2)*R(2))
=R(1)
D(2,2)*R(2) =(S(2)*R(2))*R(2)
=S(2)
00110111
↑ S(2)
D(1,1)*D(3,1)*D(3,2)=(S(1)*R(1)*R(2))*R(1)*R(2)
=S(1)
10110010
↑ S(1)
D(1,2)*D(3,1)*D(3,2)=(S(2)*R(1)*R(2))*R(1)*R(2)
=S(2)
00110111
↑ S(2)
D(2,1)*D(3,1)=(S(1)*R(1))*R(1)
=S(1)
10110010
↑ S(1)
D(2,2)*D(3,2)=(S(2)*R(2))*R(2)
=S(2)
00110111
↑ S(2)

FIG. 8

DIVISION INTO THREE (n=3)
ORIGINAL DATA CAN BE RECOVERED FROM ANY TWO DIVIDED DATA

(m IS ARBITRARY INTEGER m>0) →CONTINUED TO TAIL OF ORIGINAL DATA S

VALUE OF j	1	2	・・・	j=2×m+1	j+1	・・・
ORIGINAL DATA S(j)	S(1)	S(2)	・・・	S(j)	S(j+1)	・・・
RANDOM NUMBER R(j)	R(1)	R(2)	・・・	R(j)	R(j+1)	・・・
DIVIDED PARTIAL DATA D(1,j)	S(1)*R(1)*R(2)	S(2)*R(1)*R(2)	・・・	S(j)*R(j)*R(j+1)	S(j+1)*R(j)*R(j+1)	・・・
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)	S(2) *R(2)	・・・	S(j)*R(j)	S(j+1) *R(j+1)	・・・
DIVIDED PARTIAL DATA D(3,j)	R(1)	R(2)	・・・	R(j)	R(j+1)	・・・

FIG. 9

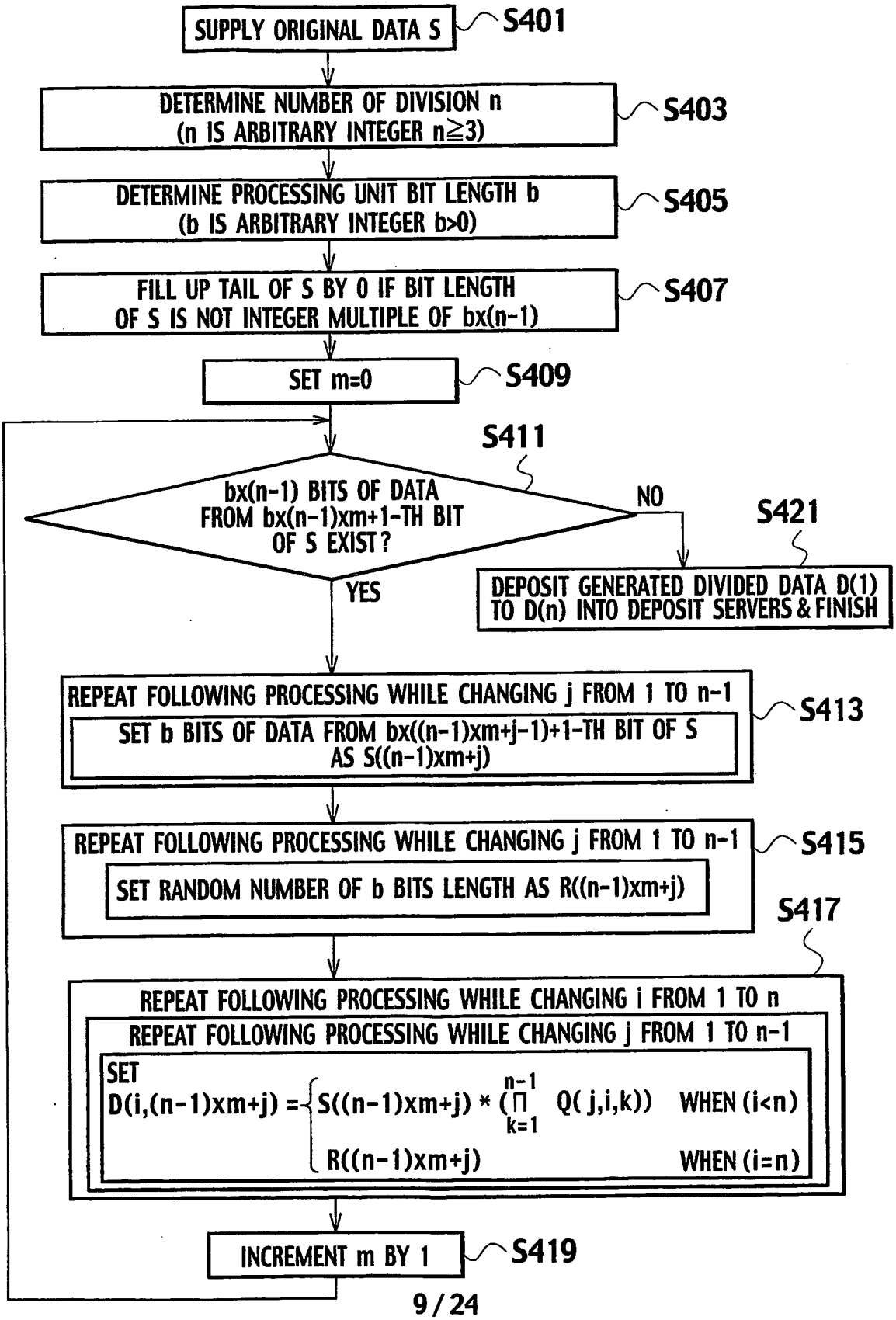
SUPPLY ORIGINAL DATA S
S401
DETERMINE NUMBER OF DIVISION n
(n IS ARBITRARY INTEGER n≧3)
S403
DETERMINE PROCESSING UNIT BIT LENGTH b
(b IS ARBITRARY INTEGER b>0)
S405
FILL UP TAIL OF S BY 0 IF BIT LENGTH
OF S IS NOT INTEGER MULTIPLE OF bx(n-1)
S407
SET m=0
S409
S411
bx(n-1) BITS OF DATA
FROM bx(n-1)xm+1-TH BIT
OF S EXIST?
NO
S421
DEPOSIT GENERATED DIVIDED DATA D(1)
TO D(n) INTO DEPOSIT SERVERS & FINISH
YES
REPEAT FOLLOWING PROCESSING WHILE CHANGING j FROM 1 TO n-1
SET b BITS OF DATA FROM bx((n-1)xm+j-1)+1-TH BIT OF S
AS S((n-1)xm+j)
S413
REPEAT FOLLOWING PROCESSING WHILE CHANGING j FROM 1 TO n-1
SET RANDOM NUMBER OF b BITS LENGTH AS R((n-1)xm+j)
S415
S417
REPEAT FOLLOWING PROCESSING WHILE CHANGING i FROM 1 TO n
REPEAT FOLLOWING PROCESSING WHILE CHANGING j FROM 1 TO n-1
SET
D(i,(n-1)xm+j) = S((n-1)xm+j) * (Π k=1 to n-1 Q(j,i,k)) WHEN (i<n)
R((n-1)xm+j) WHEN (i=n)
INCREMENT m BY 1
S419

FIG. 10

ORIGINAL

VALUE OF j	1	2	· · ·	j=2m+1	j+1	· · ·
ORIGINAL DATA S(j)	S(1)	S(2)	· · ·	S(j)	S(j+1)	· · ·
RANDOM NUMBER R(j)	R(1)	R(2)	· · ·	R(j)	R(j+1)	· · ·
DIVIDED PARTIAL DATA D(1,j)	S(1)*R(1)*R(2)	S(2)*R(1)*R(2)	· · ·	S(j)*R(j)*R(j+1)	S(j+1)*R(j)*R(j+1)	· · ·
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)	S(2) *R(2)	· · ·	S(j)*R(j)	S(j+1) *R(j+1)	· · ·
DIVIDED PARTIAL DATA D(3,j)	R(1)	R(2)	· · ·	R(j)	R(j+1)	· · ·

▽ ▽

ROTATE (1, D(1, 2), D(2, 2))

▽ ▽

ROTATE (1, D(1, j+1), D(2, j+1))

▽ ▽

IMPROVEMENT

VALUE OF j	1	2	· · ·	j=2m+1	j+1	· · ·
ORIGINAL DATA S(j)	S(1)	S(2)	· · ·	S(j)	S(j+1)	· · ·
RANDOM NUMBER R(j)	R(1)	R(2)	· · ·	R(j)	R(j+1)	· · ·
DIVIDED PARTIAL DATA D(1,j)	S(1)*R(1)*R(2)	S(2) *R(2)	· · ·	S(j)*R(j)*R(j+1)	S(j+1) *R(j+1)	· · ·
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)	S(2)*R(1)*R(2)	· · ·	S(j)*R(j)	S(j+1)*R(j)*R(j+1)	· · ·
DIVIDED PARTIAL DATA D(3,j)	R(1)	R(2)	· · ·	R(j)	R(j+1)	· · ·

FIG. 11

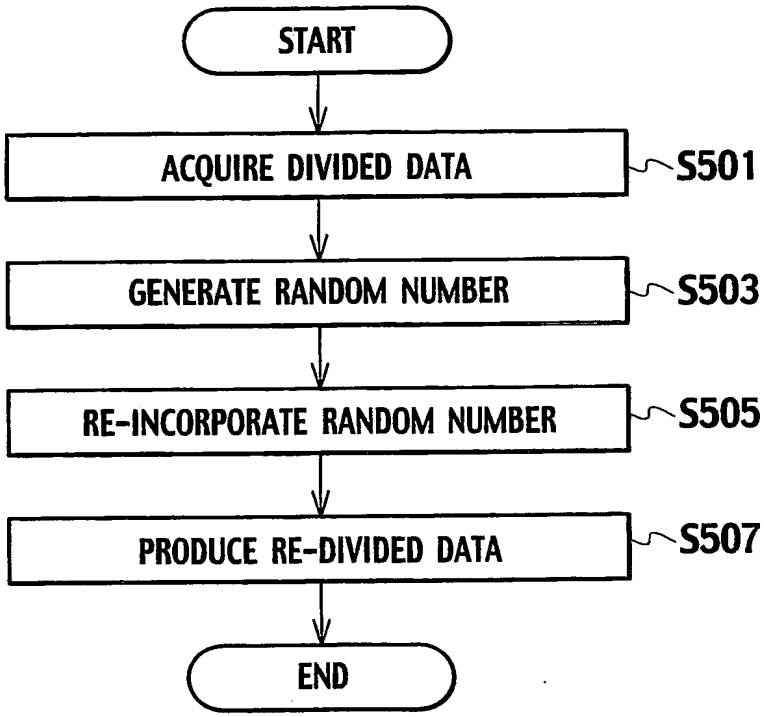
START
ACQUIRE DIVIDED DATA
S501
GENERATE RANDOM NUMBER
S503
RE-INCORPORATE RANDOM NUMBER
S505
PRODUCE RE-DIVIDED DATA
S507
END

FIG. 12

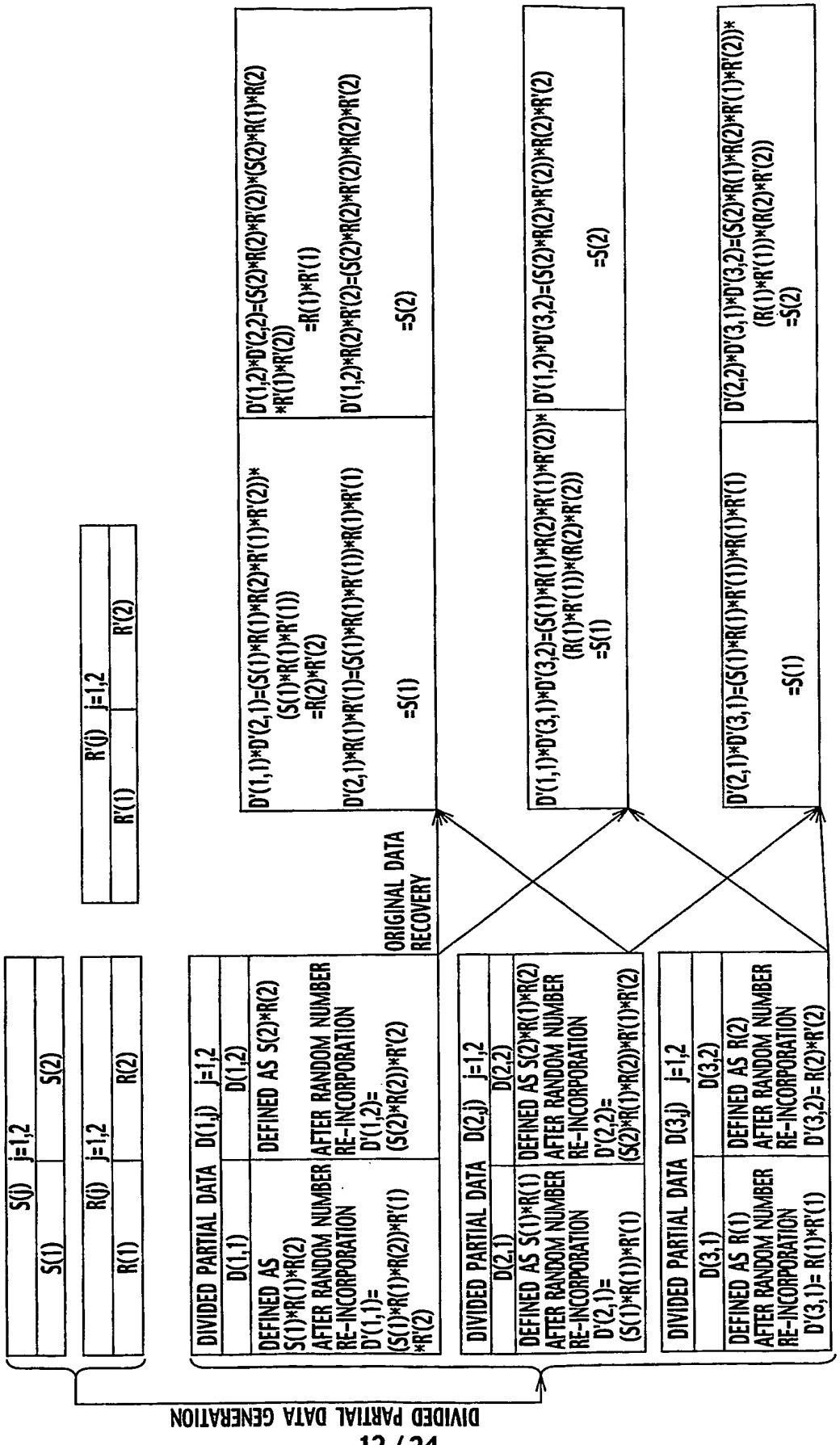
S(j) j=1,2
S(1)
S(2)
R(j) j=1,2
R(1)
R(2)
R'(j) j=1,2
R'(1)
R'(2)
DIVIDED PARTIAL DATA D(1,j) j=1,2
D(1,1)
D(1,2)
DEFINED AS S(1)*R(1)*R(2) AFTER RANDOM NUMBER RE-INCORPORATION D'(1,1)= (S(1)*R(1)*R(2))*R'(1) *R'(2)
DEFINED AS S(2)*R(2) AFTER RANDOM NUMBER RE-INCORPORATION D'(1,2)= (S(2)*R(2))*R'(2)
DIVIDED PARTIAL DATA D(2,j) j=1,2
D(2,1)
D(2,2)
DEFINED AS S(1)*R(1) AFTER RANDOM NUMBER RE-INCORPORATION D'(2,1)= (S(1)*R(1))*R'(1)
DEFINED AS S(2)*R(1)*R(2) AFTER RANDOM NUMBER RE-INCORPORATION D'(2,2)= (S(2)*R(1)*R(2))*R'(1)*R'(2)
DIVIDED PARTIAL DATA D(3,j) j=1,2
D(3,1)
D(3,2)
DEFINED AS R(1) AFTER RANDOM NUMBER RE-INCORPORATION D'(3,1)= R(1)*R'(1)
DEFINED AS R(2) AFTER RANDOM NUMBER RE-INCORPORATION D'(3,2)= R(2)*R'(2)
ORIGINAL DATA RECOVERY
D'(1,1)*D'(2,1)=(S(1)*R(1)*R(2)*R'(1)*R'(2))* (S(1)*R(1)*R'(1)) =R(2)*R'(2)
D'(2,1)*R(1)*R'(1)=(S(1)*R(1)*R'(1))*R(1)*R'(1)
=S(1)
D'(1,2)*D'(2,2)=(S(2)*R(2)*R'(2))*(S(2)*R(1)*R(2) *R'(1)*R'(2)) =R(1)*R'(1)
D'(1,2)*R(2)*R'(2)=(S(2)*R(2)*R'(2))*R(2)*R'(2)
=S(2)
D'(1,1)*D'(3,1)*D'(3,2)=(S(1)*R(1)*R(2)*R'(1)*R'(2))* (R(1)*R'(1))*(R(2)*R'(2)) =S(1)
D'(1,2)*D'(3,2)=(S(2)*R(2)*R'(2))*R(2)*R'(2)
=S(2)
D'(2,1)*D'(3,1)=(S(1)*R(1)*R'(1))*R(1)*R'(1)
=S(1)
D'(2,2)*D'(3,1)*D'(3,2)=(S(2)*R(1)*R(2)*R'(1)*R'(2))* (R(1)*R'(1))*(R(2)*R'(2)) =S(2)
DIVIDED PARTIAL DATA GENERATION

FIG. 13

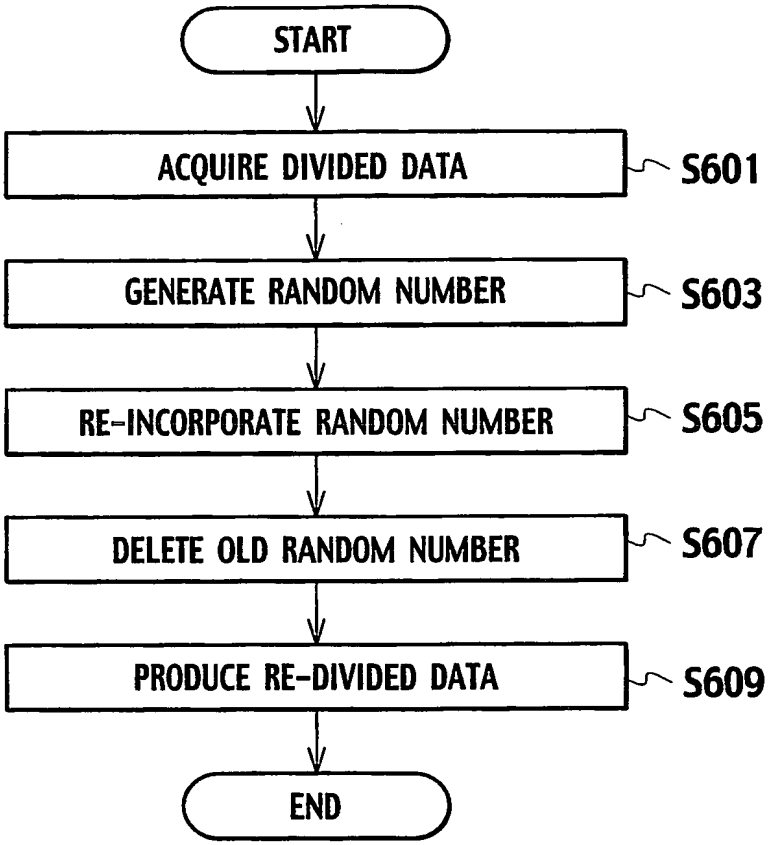
START
ACQUIRE DIVIDED DATA
S601
GENERATE RANDOM NUMBER
S603
RE-INCORPORATE RANDOM NUMBER
S605
DELETE OLD RANDOM NUMBER
S607
PRODUCE RE-DIVIDED DATA
S609
END

FIG. 14

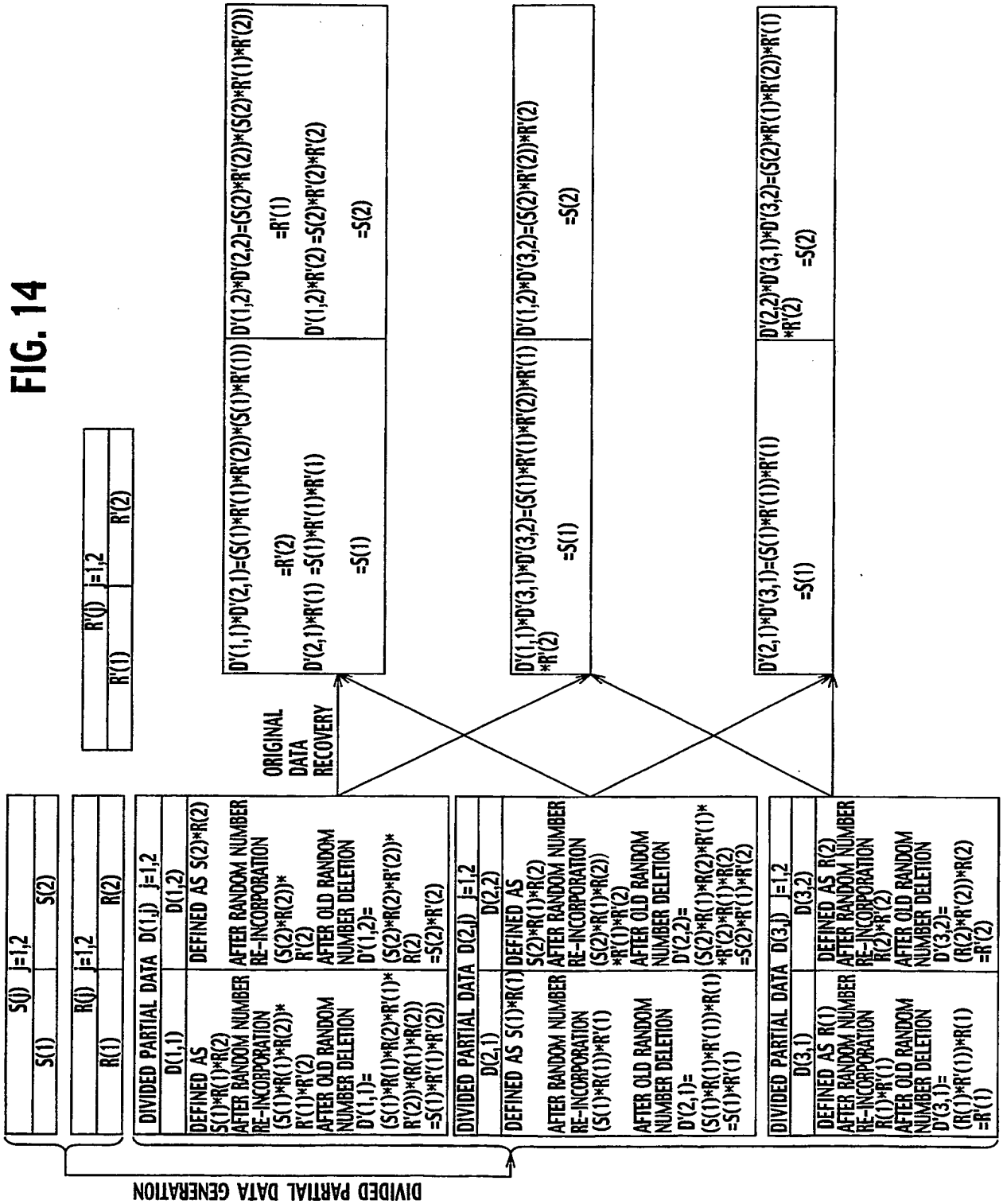
S(j) j=1,2
S(1)
S(2)
R(j) j=1,2
R(1)
R(2)
DIVIDED PARTIAL DATA GENERATION
DIVIDED PARTIAL DATA D(1,j) j=1,2
D(1,1)
D(1,2)
DEFINED AS S(1)*R(1)*R(2) AFTER RANDOM NUMBER RE-INCORPORATION (S(1)*R(1)*R(2))* R'(1)*R'(2) AFTER OLD RANDOM NUMBER DELETION D'(1,1)= (S(1)*R(1)*R(2)*R'(1)* R'(2))*(R(1)*R(2)) =S(1)*R'(1)*R'(2))
DEFINED AS S(2)*R(2) AFTER RANDOM NUMBER RE-INCORPORATION (S(2)*R(2))* R'(2) AFTER OLD RANDOM NUMBER DELETION D'(1,2)= (S(2)*R(2)*R'(2))* R(2) =S(2)*R'(2)
DIVIDED PARTIAL DATA D(2,j) j=1,2
D(2,1)
D(2,2)
DEFINED AS S(1)*R(1) AFTER RANDOM NUMBER RE-INCORPORATION (S(1)*R(1))*R'(1) AFTER OLD RANDOM NUMBER DELETION D'(2,1)= (S(1)*R(1)*R'(1))*R(1) =S(1)*R'(1)
DEFINED AS S(2)*R(1)*R(2) AFTER RANDOM NUMBER RE-INCORPORATION (S(2)*R(1)*R(2)) *R'(1)*R'(2) AFTER OLD RANDOM NUMBER DELETION D'(2,2)= (S(2)*R(1)*R(2)*R'(1)* *R'(2)*R(1)*R(2) =S(2)*R'(1)*R'(2)
DIVIDED PARTIAL DATA D(3,j) j=1,2
D(3,1)
D(3,2)
DEFINED AS R(1) AFTER RANDOM NUMBER RE-INCORPORATION R(1)*R'(1) AFTER OLD RANDOM NUMBER DELETION D'(3,1)= (R(1)*R'(1))*R(1) =R'(1)
DEFINED AS R(2) AFTER RANDOM NUMBER RE-INCORPORATION R(2)*R'(2) AFTER OLD RANDOM NUMBER DELETION D'(3,2)= (R(2)*R'(2))*R(2) =R'(2)
R'(j) j=1,2
R'(1)
R'(2)
ORIGINAL DATA RECOVERY
D'(1,1)*D'(2,1)=(S(1)*R'(1)*R'(2))*(S(1)*R'(1)) =R'(2) D'(2,1)*R'(1) =S(1)*R'(1)*R'(1) =S(1)
D'(1,2)*D'(2,2)=(S(2)*R'(2))*(S(2)*R'(1)*R'(2)) =R'(1) D'(1,2)*R'(2) =S(2)*R'(2)*R'(2) =S(2)
D'(1,1)*D'(3,1)*D'(3,2)=(S(1)*R'(1)*R'(2))*R'(1) *R'(2) =S(1)
D'(1,2)*D'(3,2)=(S(2)*R'(2))*R'(2) =S(2)
D'(2,1)*D'(3,1)=(S(1)*R'(1))*R'(1) =S(1)
D'(2,2)*D'(3,1)*D'(3,2)=(S(2)*R'(1)*R'(2))*R'(1) *R'(2) =S(2)

FIG. 15

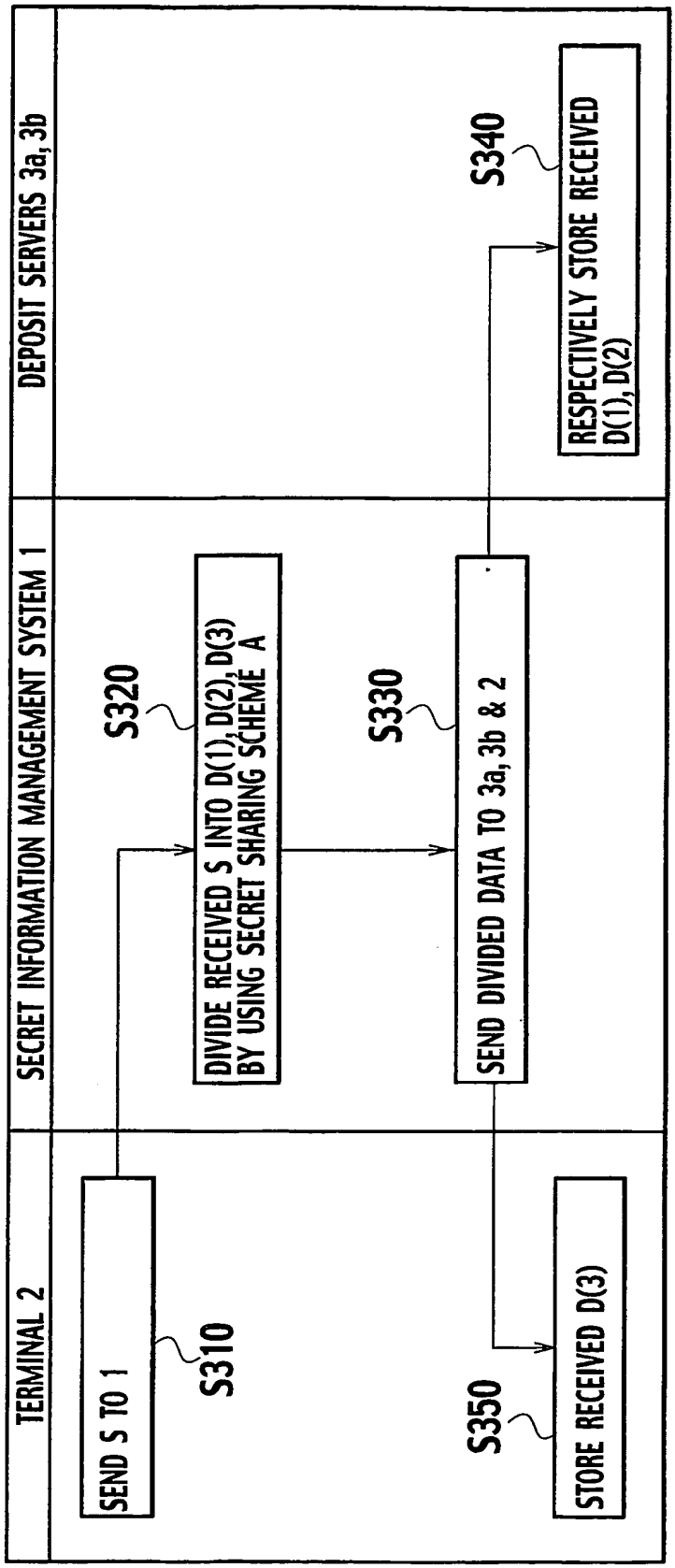
TERMINAL 2
SECRET INFORMATION MANAGEMENT SYSTEM 1
DEPOSIT SERVERS 3a, 3b
SEND S TO 1
S310
DIVIDE RECEIVED S INTO D(1), D(2), D(3) BY USING SECRET SHARING SCHEME A
S320
SEND DIVIDED DATA TO 3a, 3b & 2
S330
RESPECTIVELY STORE RECEIVED D(1), D(2)
S340
STORE RECEIVED D(3)
S350

FIG. 16

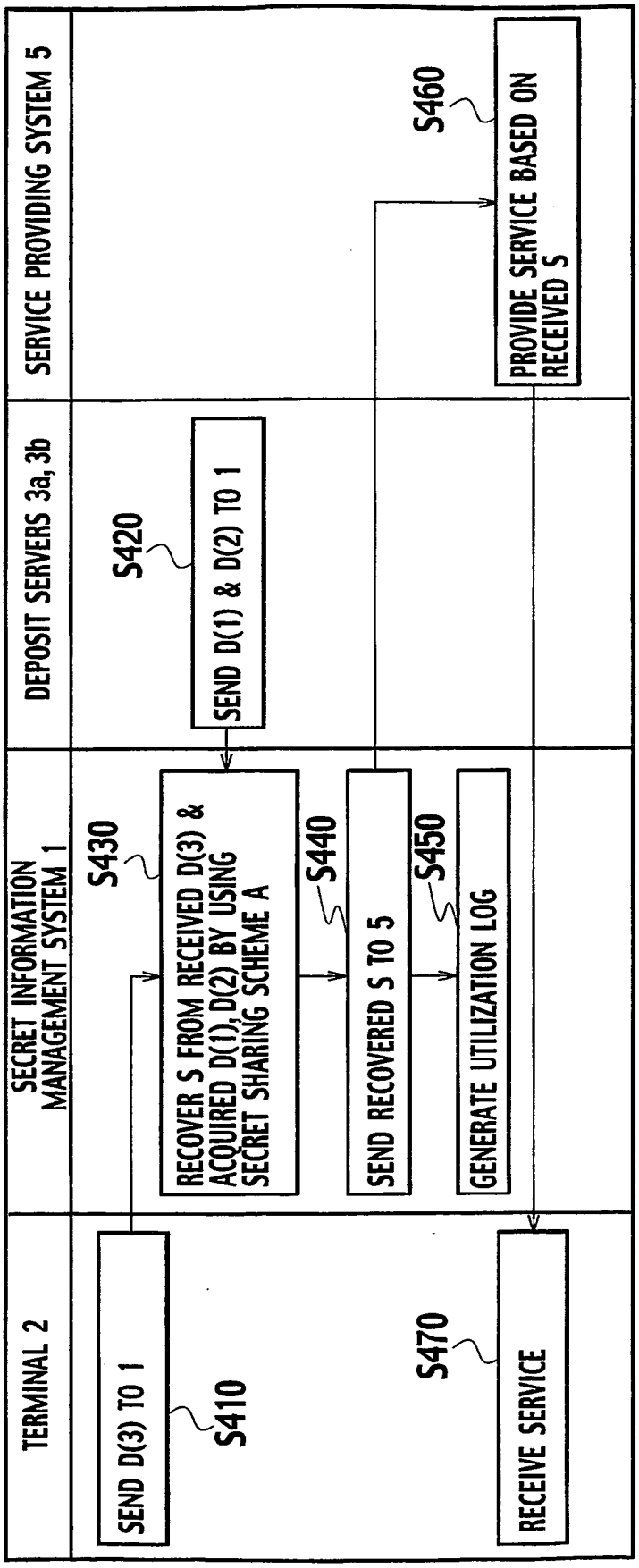
TERMINAL 2
SECRET INFORMATION MANAGEMENT SYSTEM 1
DEPOSIT SERVERS 3a, 3b
SERVICE PROVIDING SYSTEM 5
SEND D(3) TO 1
S410
SEND D(1) & D(2) TO 1
S420
RECOVER S FROM RECEIVED D(3) & ACQUIRED D(1), D(2) BY USING SECRET SHARING SCHEME A
S430
SEND RECOVERED S TO 5
S440
GENERATE UTILIZATION LOG
S450
PROVIDE SERVICE BASED ON RECEIVED S
S460
RECEIVE SERVICE
S470

FIG. 17

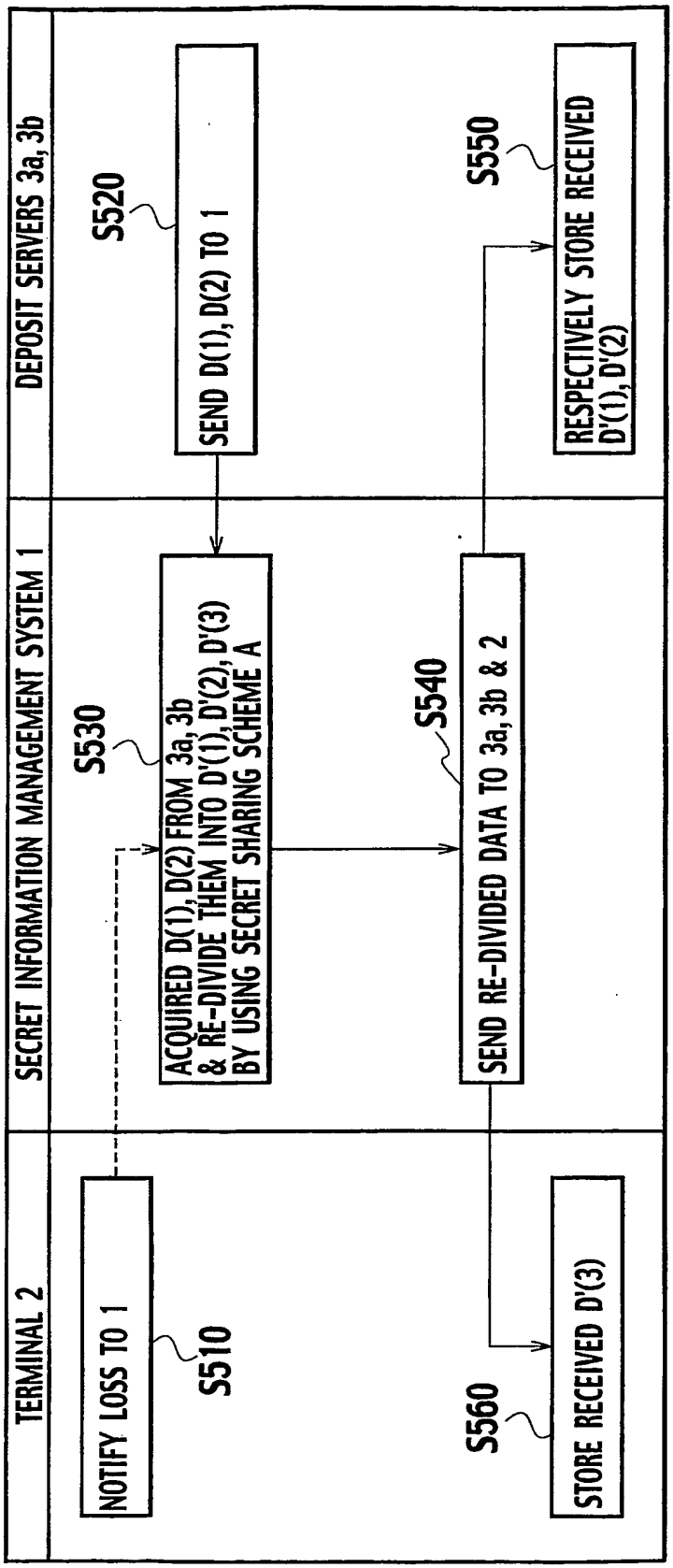
TERMINAL 2
SECRET INFORMATION MANAGEMENT SYSTEM 1
DEPOSIT SERVERS 3a, 3b
NOTIFY LOSS TO 1
S510
S520
SEND D(1), D(2) TO 1
S530
ACQUIRED D(1), D(2) FROM 3a, 3b
& RE-DIVIDE THEM INTO D'(1), D'(2), D'(3)
BY USING SECRET SHARING SCHEME A
S540
SEND RE-DIVIDED DATA TO 3a, 3b & 2
S550
RESPECTIVELY STORE RECEIVED
D'(1), D'(2)
S560
STORE RECEIVED D'(3)

FIG. 18

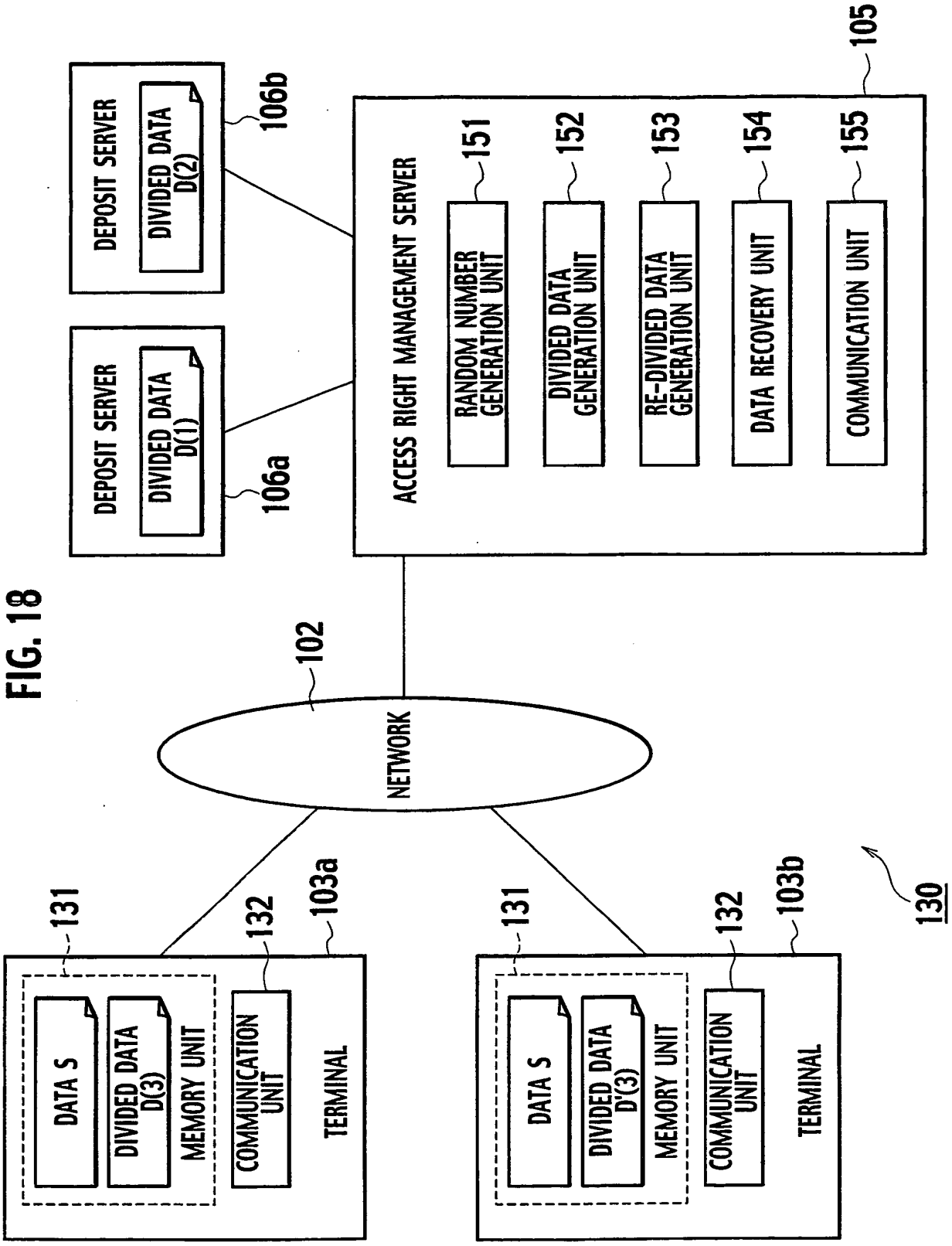
DEPOSIT SERVER
DIVIDED DATA D(2)
106b
DEPOSIT SERVER
DIVIDED DATA D(1)
106a
ACCESS RIGHT MANAGEMENT SERVER
105
RANDOM NUMBER GENERATION UNIT
151
DIVIDED DATA GENERATION UNIT
152
RE-DIVIDED DATA GENERATION UNIT
153
DATA RECOVERY UNIT
154
COMMUNICATION UNIT
155
NETWORK
102
130
TERMINAL
103a
131
DATA S
DIVIDED DATA D(3)
MEMORY UNIT
132
COMMUNICATION UNIT
TERMINAL
103b
131
DATA S
DIVIDED DATA D'(3)
MEMORY UNIT
132
COMMUNICATION UNIT

FIG. 19

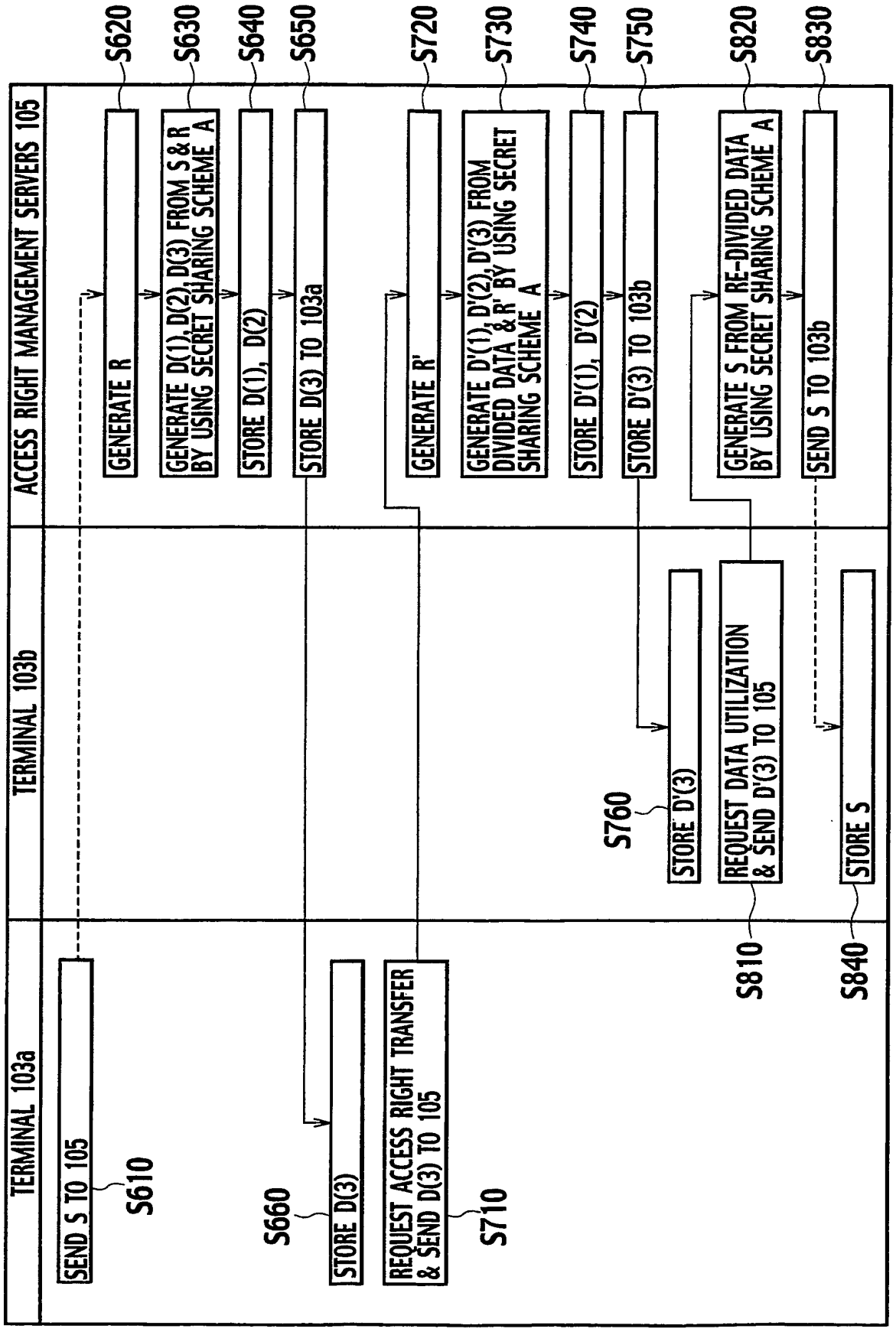
ACCESS RIGHT MANAGEMENT SERVERS 105
TERMINAL 103b
TERMINAL 103a
S620
GENERATE R
S630
GENERATE D(1), D(2), D(3) FROM S & R BY USING SECRET SHARING SCHEME A
S640
STORE D(1), D(2)
S650
STORE D(3) TO 103a
S720
GENERATE R'
S730
GENERATE D'(1), D'(2), D'(3) FROM DIVIDED DATA & R' BY USING SECRET SHARING SCHEME A
S740
STORE D'(1), D'(2)
S750
STORE D'(3) TO 103b
S820
GENERATE S FROM RE-DIVIDED DATA BY USING SECRET SHARING SCHEME A
S830
SEND S TO 103b
S610
SEND S TO 105
S660
STORE D(3)
S710
REQUEST ACCESS RIGHT TRANSFER & SEND D(3) TO 105
S760
STORE D'(3)
S810
REQUEST DATA UTILIZATION & SEND D'(3) TO 105
S840
STORE S

FIG. 20

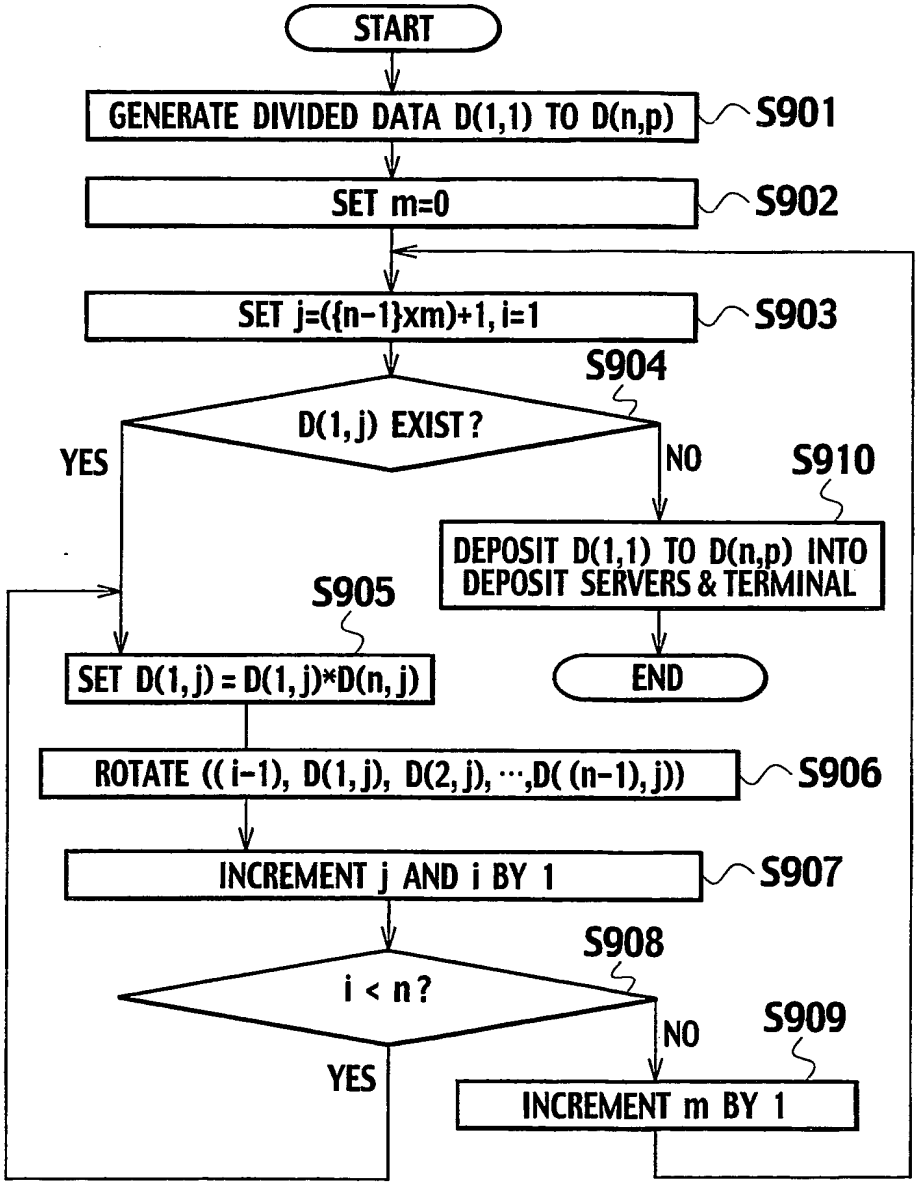
START
GENERATE DIVIDED DATA D(1,1) TO D(n,p)
S901
SET m=0
S902
SET j=({n-1}xm)+1, i=1
S903
S904
D(1, j) EXIST?
YES
NO
S910
DEPOSIT D(1,1) TO D(n,p) INTO DEPOSIT SERVERS & TERMINAL
END
S905
SET D(1, j) = D(1, j)*D(n, j)
ROTATE ((i-1), D(1, j), D(2, j), ···,D((n-1), j))
S906
INCREMENT j AND i BY 1
S907
S908
i < n?
NO
S909
YES
INCREMENT m BY 1

FIG. 21

ORIGINAL

VALUE OF j	1	2	3	. . .
ORIGINAL DATA S(j)	S(1)	S(2)	S(3)	. . .
RANDOM NUMBER R(j)	R(1)	R(2)	R(3)	. . .
DIVIDED PARTIAL DATA D(1,j)	S(1)*R(1)*R(2)*R(3)	S(2)*R(1)*R(2)*R(3)	S(3)*R(1)*R(2)*R(3)	. . .
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)*R(2)	S(2) *R(2)*R(3)	S(3)*R(1) *R(3)	. . .
DIVIDED PARTIAL DATA D(3,j)	S(1)*R(1)	S(2) *R(2)	S(3) *R(3)	. . .
DIVIDED PARTIAL DATA D(4,j)	R(1)	R(2)	R(3)	

▽ ROTATE (1, D(1, 2), D(2, 2), D(3, 2)) ▽ ROTATE (2, D(1, 3), D(2, 3), D(3, 3)) ▽

IMPROVEMENT

VALUE OF j	1	2	3	. . .
ORIGINAL DATA S(j)	S(1)	S(2)	S(3)	. . .
RANDOM NUMBER R(j)	R(1)	R(2)	R(3)	. . .
DIVIDED PARTIAL DATA D(1,j)	S(1)~~*R(1)~~*R(2)*R(3)	S(2) *R(2)	S(3)*R(1) *R(3)	. . .
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)*R(2)	S(2)*R(1)~~*R(2)~~*R(3)	S(3) *R(3)	. . .
DIVIDED PARTIAL DATA D(3,j)	S(1)*R(1)	S(2) *R(2)*R(3)	S(3)*R(1)*R(2)~~*R(3)~~	. . .
DIVIDED PARTIAL DATA D(4,j)	R(1)	R(2)	R(3)	

FIG. 22

ORIGINAL

. . .	j=3m+1	j+1	j+2	. . .
. . .	S(j)	S(j+1)	S(j+2)	. . .
. . .	R(j)	R(j+1)	R(j+2)	. . .
. . .	S(j)*R(j)*R(j+1)*R(j+2)	S(j+1)*R(j)*R(j+1)*R(j+2)	S(j+2)*R(j)*R(j+1)*R(j+2)	. . .
. . .	S(j)*R(j)*R(j+1)	S(j+1) *R(j+1)*R(j+2)	S(j+2)*R(j) *R(j+2)	. . .
. . .	S(j)*R(j)	S(j+1) *R(j+1)	S(j+1) *R(j+2)	
. . .	R(j)	*R(j+1)	*R(j+2)	. . .

▽ ▽

ROTATE (1, D(1, j+1), D(2, j+1), D(3, j+1))

ROTATE (2, D(1, j+2), D(2, j+2), D(3, j+2))

▽ ▽

IMPROVEMENT

. . .	j=3m+1	j+1	j+2	. . .
. . .	S(j)	S(j+1)	S(j+2)	. . .
. . .	R(j)	R(j+1)	R(j+2)	. . .
. . .	S(j)~~*R(j)~~*R(j+1)*R(j+2)	S(j+1) *R(j+1)	S(j+2)*R(j) *R(j+2)	. . .
. . .	S(j)*R(j)*R(j+1)	S(j+1)*R(j)~~*R(j+1)~~*R(j+2)	S(j+2) *R(j+2)	. . .
. . .	S(j)*R(j)	S(j+1) *R(j+1)*R(j+2)	S(j+2)*R(j)*R(j+1)~~*R(j+2)~~	
. . .	R(j)	*R(j+1)	*R(j+2)	. . .

FIG. 23

ORIGINAL

VALUE OF j	1	2	3	4	...
ORIGINAL DATA S(j)	S(1)	S(2)	S(3)		...
RANDOM NUMBER R(j)	R(1)	R(2)	R(3)		...
DIVIDED PARTIAL DATA D(1,j)	S(1)*R(1)*R(2)*R(3)*R(4)	S(2)*R(1)*R(2)*R(3)*R(4)	S(3)*R(1)*R(2)*R(3)*R(4)	S(4)*R(1)*R(2)*R(3)*R(4)	...
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)*R(2)*R(3)	S(2) *R(2)*R(3)*R(4)	S(3)*R(1) *R(3)*R(4)	S(4)*R(1)*R(2) *R(4)	...
DIVIDED PARTIAL DATA D(3,j)	S(1)*R(1)*R(2)	S(2) *R(2)*R(3)	S(3) *R(3)*R(4)	S(4)*R(1) *R(4)	...
DIVIDED PARTIAL DATA D(4,j)	S(1)*R(1)	S(2) *R(2)	S(3) *R(3)	S(4) *R(4)	...
DIVIDED PARTIAL DATA D(5,j)	R(1)	*R(2)	*R(3)	*R(4)	...

ROTATE (1, D(1, 2), D(2, 2), (3, 2), D(4, 2))

ROTATE (2, D(1, 3), D(2, 3), (3, 3), D(4, 3))

ROTATE (3, D(1, 4), D(2, 4), (3, 4), D(4, 4))

IMPROVEMENT

VALUE OF j	1	2	3	4	...
ORIGINAL DATA S(j)	S(1)	S(2)	S(3)		...
RANDOM NUMBER R(j)	R(1)	R(2)	R(3)		...
DIVIDED PARTIAL DATA D(1,j)	S(1)~~*R(1)~~*R(2)*R(3)*R(4)	S(2) *R(2)	S(3) *R(3)*R(4)	S(4)*R(1)*R(2) *R(4)	...
DIVIDED PARTIAL DATA D(2,j)	S(1)*R(1)*R(2)*R(3)	S(2)*R(1)~~*R(2)~~*R(3)*R(4)	S(3) *R(3)	S(4)*R(1) *R(4)	...
DIVIDED PARTIAL DATA D(3,j)	S(1)*R(1)*R(2)	S(2) *R(2)*R(3)*R(4)	S(3)*R(1)*R(2)~~*R(3)~~*R(4)	S(4) *R(4)	...
DIVIDED PARTIAL DATA D(4,j)	S(1)*R(1)	S(2) *R(2)*R(3)	S(3)*R(1) *R(3)*R(4)	S(4)*R(1)*R(2)*R(3)~~*R(4)~~	...
DIVIDED PARTIAL DATA D(5,j)	R(1)	*R(2)	*R(3)	*R(4)	...

FIG. 24

ORIGINAL

・・・	j=4m+1	j+1	j+2	j+3	・・・
・・・	S(j)	S(j+1)	S(j+2)	S(j+3)	・・・
・・・	R(j)	R(j+1)	R(j+2)	R(j+3)	・・・
・・・	S(j)*R(j)*R(j+1)*R(j+2)*R(j+3)	S(j+1)*R(j)*R(j+1)*R(j+2)*R(j+3)	S(j+2)*R(j)*R(j+1)*R(j+2)*R(j+3)	S(j+3)*R(j)*R(j+1)*R(j+2)*R(j+3)	・・・
・・・	S(j)*R(j)*R(j+1)*R(j+2)	S(j+1) *R(j+1)*R(j+2)*R(j+3)	S(j+2)*R(j) *R(j+2)*R(j+3)	S(j+3)*R(j)*R(j+1) *R(j+3)	・・・
・・・	S(j)*R(j)*R(j+1)	S(j+1) *R(j+1)*R(j+2)	S(j+2) *R(j+2)*R(j+3)	S(j+3)*R(j) *R(j+3)	・・・
・・・	S(j)*R(j)	S(j+1) *R(j+1)	S(j+2) *R(j+2)	S(j+3) *R(j+3)	・・・
・・・	R(j)	R(j+1)	R(j+2)	R(j+3)	・・・

ROTATE (1, D(1, j+1), D(2, j+1), D(3, j+1), D(4, j+1))

ROTATE (2, D(1, j+2), D(2, j+2), D(3, j+2), D(4, j+2))

ROTATE (3, D(1, j+3), D(2, j+3), D(3, j+3), D(4, j+3))

IMPROVEMENT

・・・	j=4m+1	j+1	j+2	j+3	・・・
・・・	S(j)	S(j+1)	S(j+2)	S(j+3)	・・・
・・・	R(j)	R(j+1)	R(j+2)	R(j+3)	・・・
・・・	S(j)~~*R(j)~~*R(j+1)*R(j+2)*R(j+3)	S(j+1) *R(j+1)	S(j+1) *R(j+2)*R(j+3)	S(j+3)*R(j)*R(j+1) *R(j+3)	・・・
・・・	S(j)*R(j)*R(j+1)*R(j+2)	S(j+1)*R(j)*~~R(j+1)~~*R(j+2)*R(j+3)	S(j+2) *R(j+2)	S(j+3)*R(j) *R(j+3)	・・・
・・・	S(j)*R(j)*R(j+1)	S(j+1) *R(j+1)*R(j+2)*R(j+3)	S(j+2)*R(j)*R(j+1)*~~R(j+2)~~*R(j+3)	S(j+3) *R(j+3)	・・・
・・・	S(j)*R(j)	S(j+1) *R(j+1)*R(j+2)	S(j+2)*R(j) *R(j+2)*R(j+3)	S(j+3)*R(j)*R(j+1)*R(j+2)*~~R(j+3)~~	・・・
・・・	R(j)	R(j+1)	R(j+2)	R(j+3)	・・・